Optimal Design of Complex Systems Using the Entero Code System

Paul N. Demmie[1]

Summary

This paper summarizes an approach to optimal design of complex systems that is being implemented in the *Entero Code System*. *Entero* employs the *DAKOTA* iterator toolkit for the optimization and uncertainty analysis. The *Entero Code System*, its methods, and the *DAKOTA* toolkit are presented along with several examples using the *SAMPLL* code to simulate penetrator performance and evaluate objective functions.

Introduction

The design of complex systems may involve optimization of critical system parameters or performance with constraints imposed by design requirements or the environment. Also, there may be uncertainties in the design due to tolerances in system features or knowledge of the environment. Engineers often lack an integrated framework to perform the analyses needed for designing such systems.

The *Entero Code System* [1] is being developed to provide an integrated framework for the design of complex systems. Figure 1 states the goal of the *Entero Code System*.

To develop a flexible environment for the design and analysis of complex systems in diverse environments from a system-oriented, module-oriented point of view using models with varying degrees of fidelity and accessed by means of a web-like user interface.

Figure 1. Goal of the Entero Code System.

Figure 1 shows that the goal of the *Entero Code System* includes several key features. First, *Entero* is not a monolithic computer program, but a flexible environment, which through its integrator calculates the physical effects requested by the user. Its architecture is intended to be extensible to permit inclusion of new physical phenomena and services in the code system. Furthermore, *Entero* is system and module oriented. It treats a complex system as a set of interacting modules. This view emphasizes the importance of module interactions. System engineers deal first with physical modules rather than the computational models used to represent them. *Entero* will support a range of models from low-fidelity, parametric

[1] Sandia National Laboratories, Albuquerque, NM, 87185-0820, USA. Sandia National Laboratories is a multiprogram laboratory operated by the Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

models to higher fidelity three-dimensional, multi-physics models. For certain physical phenomena, heat transfer for example, different levels of fidelity can be used for the modules.

To assist in the design of a system, *Entero* provides tools to determine uncertainties in system performance due to variability in environmental conditions, material properties, parameter specifications, or modeling assumptions. It also provides tools to optimize the design of a system.

The *Entero Code System* can be accessed on the internal web at Sandia National Laboratories (SNL) through the SaaS (Simulation as a Service) homepage. It employs a web-like user interface. Although it is launched using personal computers (PCs) or workstations, it can execute its computational modules on a range of platforms from workstations to Cplant™. Cplant™ is a cluster computing system at SNL that contains many linked commercial processors. The SI/PDO (Simulation-Internet/Product-Data-Object) provides the services for communication between the client, server, computational platforms, and file servers.

Although *Entero* includes modules for diverse physical phenomena, this paper addresses only its application to the optimal design of penetrators. In the next section, the *Entero Penetrator Toolkit* is discussed. This discussion is followed by a summary of a method for optimal penetrator design that is being implemented in *Entero*. Finally, I provide several example problems to illustrate the methodology.

The Entero Penetrator Toolkit

An application of *Entero* is the integrated environment for the design and analysis of penetrators including analysis tools of different fidelities. We call this environment the *Entero Penetrator Toolkit*. Figure 2 illustrates the top-level architecture of this toolkit. This figure shows an engineering interface, the *Entero* integrator, analysis codes of increasing fidelity, computation platforms, and services. Security, access to the engineering interface, and communication between the analysis codes, services, and computational platforms are handled by the SI/PDO. After launching *Entero* and logging onto the SI/PDO, the user accesses components in the toolkit through the *Entero Engineering Interface*. Once choices of analysis code and services are made, the *Entero Integrator* controls execution on a selected computational platform. The analysis codes have different fidelities in modeling the penetrator and target modules and the physics of their interactions.

The simplest analysis code used in the penetrator toolkit is the *SAMPLL* (Simplified Analytical Model of Penetration with Lateral Loading) computer code [2].

The *Cavity Expansion (CE)* code [3] is the next level of fidelity in the *Entero Penetrator Toolkit*. Specification of the penetrator and target in *CE* is more detailed than in *SAMPLL*. Further increase in fidelity from *CE* is through its use of conservation laws to model penetrator performance and penetrator-target interactions.

The *Pronto/CE* [4] code represents the next level of fidelity in the toolkit. *Pronto* is a three-dimensional, transient, solid dynamics code for analyzing large deformations of highly nonlinear materials that are subjected to extremely high strain rates. This Lagrangian finite-element code uses an explicit-time-integration operator to integrate the equations of motion. Although the penetrator is modeled in great detail using *Pronto*, the target and physics of interaction with the target use the same methodology as the *CE* code.

The penetrator-analysis code with the highest fidelity is the *Zapotec* code [5]. *Zapotec* is a Lagrangian-Eulerian code that is built upon the technology of *Pronto* and the Eulerian *CTH* codes. *CTH* [6] is a three-dimensional, transient, solid mechanics code system that can simulate a wide range of shock-wave propagation and material-motion phenomena. It uses explicit, second-order-accurate, finite-difference numerical methods to solve the conservation equations. *Zapotec* models the penetrator using *Pronto* and the target using *CTH*. Therefore, the increase in fidelity over *Pronto*/CE is in the modeling of the target and first-principle based penetrator-target interactions.

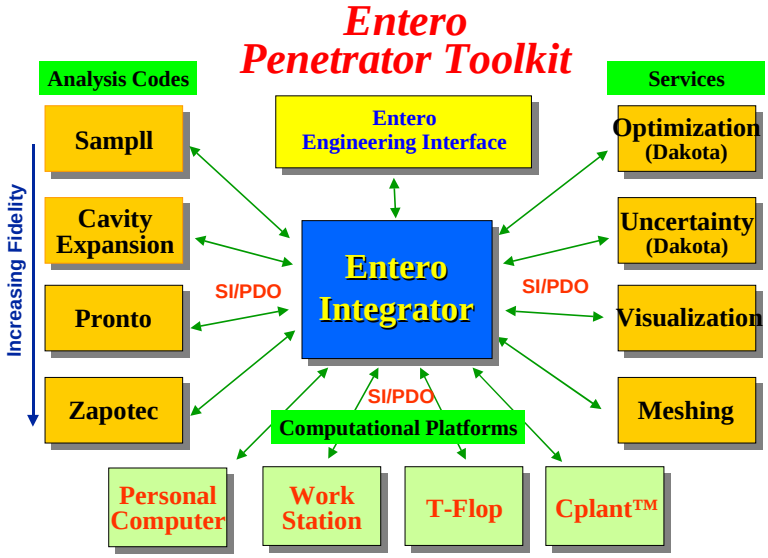

Figure 2. Top-Level Architecture of the Entero Penetrator Toolkit.

The right column of Figure 2 shows various services implemented or planned for the penetrator toolkit. Although the list is not exhaustive, it features the *DAKOTA* [7] (Design Analysis Kit for OpTimizAtion) toolkit for the optimization and uncertainty analysis. Visualization and meshing tools are planned to be included in the toolkit.

Finally, Figure 2 shows various computational platforms available for use by *Entero*. These platforms include PCs, workstations, the ASCI (Accelerated Strategic Computing Initiative) red computer (T-Flop in the Figure 2), and Cplant™.

A Method for Optimal Penetrator Design

For optimal penetrator design, *Entero* features the *DAKOTA* iterator toolkit to perform optimization and uncertainty analysis. *DAKOTA* is a flexible, extensible interface between simulation codes and analysis methods. It implements optimization with a variety of methods and uncertainty quantification with non-deterministic propagation methods, parameter estimation with nonlinear least squares solution methods, and sensitivity analysis with general-purpose parameter study capabilities. Figure 3 illustrates the *DAKOTA* framework.

DAKOTA provides a generic interface for mapping a set of parameters (design variables) into a set of responses (objective function and constraints). The application interface hides the

complexities of a given problem from the iterator (optimization or uncertainty) method. The application's details are encapsulated within the application interface that contains three main components. The input-filter program transforms the set of *DAKOTA* input parameters into input required by the simulation program. The simulation program reads its input and computes its results. Finally, the output-filter program transforms simulation results into the desired response data set. The two filter programs are generally application specific. The user supplies a *DAKOTA* input file that contains details about the parameters, including their ranges, and information about the iterator method chosen from the methods available. For the present applications, a script file controls the application interface.

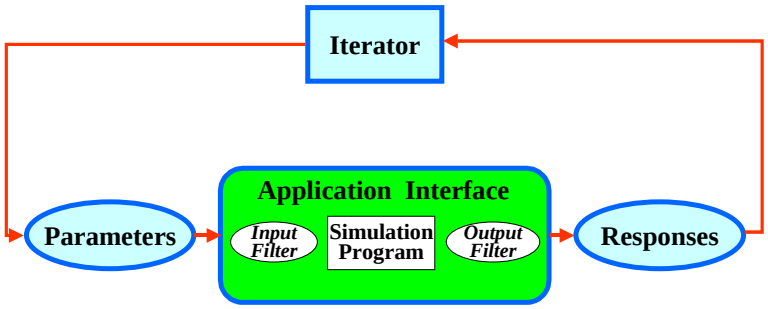

Figure 3. DAKOTA Framework.

To illustrate the *Entero* methodology with a simple analysis code, the *SAMPLL* computer code is used to simulate penetrator-target interaction. *SAMPLL* was developed in 1984 to predict penetrator-target interactions efficiently. It uses empirically based algorithms formulated from an extensive experimental base to model penetrator performance and penetrator-target interactions. During the interaction, the resistance of the target material imparts both lateral and axial loads on the penetrator that change the penetrator's motion.

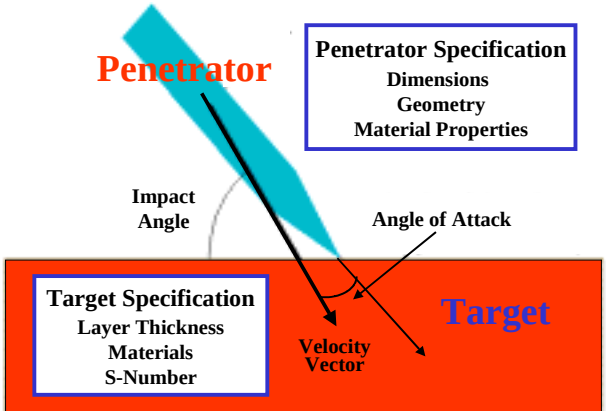

Figure 4. Penetrator Impacting a Target.

Figure 4 shows a penetrator impacting a target. This figure defines the impact angle and angle of attack. It indicates that various dimensions, geometrical features, and material properties must be specified for the penetrator and thickness, material, and S-number must be specified for the target. The S-number is a measure of the penetrability of a target.

Using *SAMPLL* as the simulation program with *DAKOTA* required that several modifications be made to *SAMPLL*. The original user interface was removed so that *SAMPLL* reads an input file that the input filter creates. Data in this file are made consistent when the input filter changes dimensions or geometry. Therefore, the density of the penetrator materials must be specified or the defaults accepted. *SAMPLL* writes an additional output file used by the output filter to provide the values of the objective and constraint functions to the iterator. Also, although not required by *DAKOTA*, an option to use an acceleration-based failure criterion was implemented. This capability makes *SAMPLL*'s failure criterion consistent with the criteria used by designers.

Some Example Problems that Illustrate the Methodology

Table 1 lists the principal input parameters for the penetrators and targets discussed in the following examples. This table does not include *SAMPLL* control parameters since they are not needed to understand the methodology. It also does not include quantities, such as center of gravity and moments of inertia, that *SAMPLL* calculates internally or are not used for the examples. For the initial velocity, impact angle, and failure criterion listed, the penetrator fails when it reaches a depth of about 2.1 feet where the magnitude of the acceleration at the center of mass exceeds the 1000-g failure criterion. In the following, four example problems are summarized to illustrate the methodology for optimal design of complex systems that is being implemented in the *Entero Code System.*

Table 1. Principal Input Parameters for Example Problems

Penetrator Data			
Total Weight	591 lb.	Failure Criterion	1000 g
Length	6.2 ft		
Number Segments	2	**Target Data**	
Diameter	13.3 in	Target Layers	1
Length		Material	soil
Segment 1	2.5 ft	S-number	5
Segment 2	3.7 ft		
Density	733 lb/ft^3	**Impact Conditions**	
Wall Thickness	0.5 in	Impact Speed	1400 ft/s
Wall Weight	400 lb.	Impact Angle	73^o
Type of Nose	ogive	Attack Angle	0

Problem 1: Determine the impact angle at which the penetration depth is a maximum.

This simple problem is valuable since the answer is known and obtaining the correct answer provides some confidence in the method. The input was appropriate to determine a solution between 0^o and 100^o using a genetic optimization algorithm. The solution obtained is 90^o. The penetrator does not fail when impacting the target at this angle and reaches a depth of 32.2 ft.

Problem 2: Suppose that constraints on the payload of the penetrator restrict the wall thickness to be less than 0.72 in. Determine the maximum penetration depth.

The input was appropriate to determine a solution for wall thickness between 0.48 in and 0.72 in. using a genetic optimization algorithm. The solution obtained is 36.6 ft. This solution is obtained for a wall thickness of 0.71 in.

Problem 3: Determine variability in penetration depth given the following. All variables are normally distributed. The means are the penetrator's wall thickness from *Problem 2*, its default strength properties, and the failure criterion, impact speed, impact and attack angles, and the target's S-number that are listed in Table 1. The standard deviations are 10% except for attack angle, which has a standard deviation of 2^{o}.

The input was appropriate to perform an uncertainty calculation using Latin hypercube sampling. The mean value for penetration depth is 19.4 ft and the standard deviation is 1.6 ft. In the simulations for the 100 sample cases, there are 52 failures, indicating that the stated variations are unacceptably large for a penetrator design or that the design is not robust.

Problem 4: Suppose that the penetrator must reach a depth of at least 10 ft to accomplish its mission. Determine the minimum wall thickness needed to accomplish its mission.

The input was appropriate to determine a minimum wall thickness subject to the constraint that the penetrator reaches a depth of at least 10 ft using a Newton optimization algorithm. The solution is 0.539 in. The penetration depth is 32.2 ft.

References

1 Gardner, D. R., Castro, J. P., Hennigan, G. L. and Cole, B. H. (2000): *A Prototype of the Entero System Engineering Code Package*, SAND2000-1368, Sandia National Laboratories.

2 Young, C. W. (1998): *Simplified Analytical Model of Penetration with Lateral Loading User's Guide*, SAND98-0978, Sandia National Laboratories.

3 Forrestal, M. J. and Tzou, D. Y. (1997): "A Spherical Cavity-Expansion Penetration Model for Concrete Targets," *Int. J. Solids Structures*, Vol. 34, pp. 4127-4146.

4 Warren, T. L. and Tabbara, M. R. (1997): *Spherical Cavity-Expansion Forcing Function in PRONTO 3D for Application to Penetration Problems*. SAND97-1174, Sandia National Laboratories.

5 Konteras, J. R., Prentice, J. K., and Attaway, S. W (1997): *A Parallel Code for Performing Coupled Eulerian/Lagrangian Solid Dynamics*, SAND97-1651A, Sandia National Laboratories.

6 McGlaun, J. M. and Thompson, S. L. (1990): "CTH: A Three-Dimensional Shock Physics Code," *International Journal of Impact Engineering*, Vol . 10, pp. 351-360.

7 Eldred, M. S., Bohnhoff, W. J., and Hart, W. E. (1999): *DAKOTA, A Multilevel Parallel Object-Oriented Framework of Design Optimization, Parameter Estimation, Sensitivity Analysis, and Uncertainty Quantification*, SAND99-0000, Sandia National Laboratories.